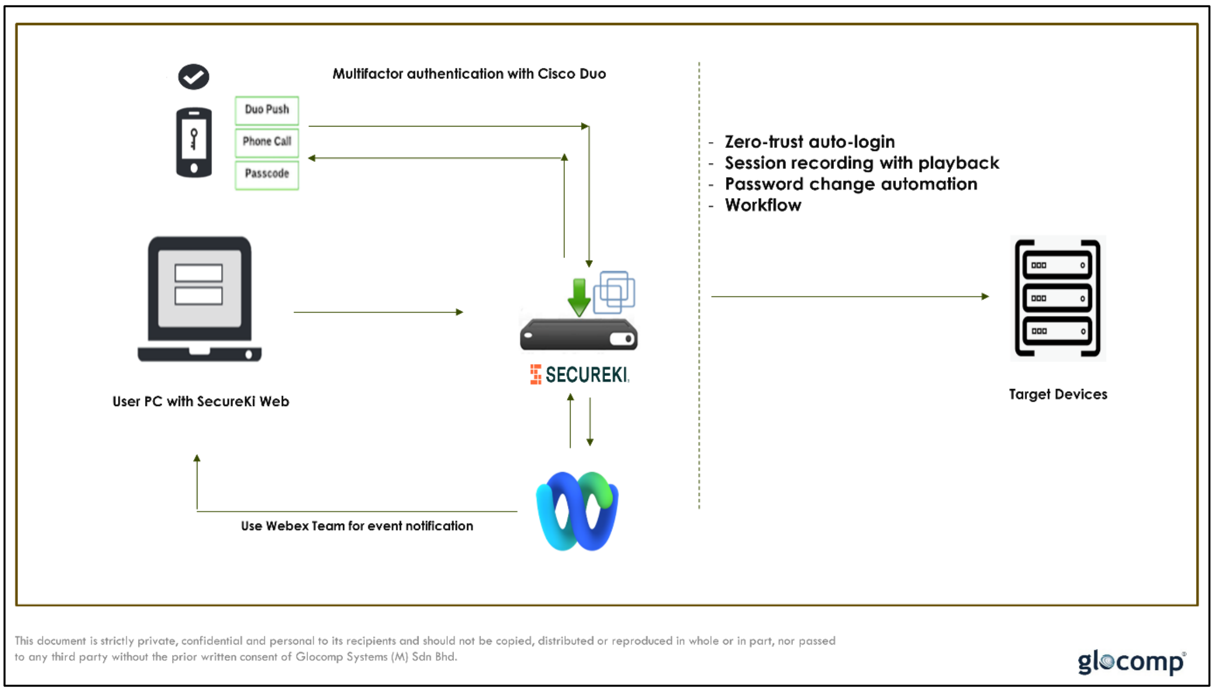
Secureki - Cisco Duo & Webex Teams Integration

Technical Specifications

Company Name	: Glocomp Systems (M) Sdn Bhd
Application Name	: SecureKi

Contents

Overview


Multifactor authentication with Cisco Duo
Duo Push
Phone Call
Passcode
- Zero-trust auto-login
- Session recording with playback
- Password change automation
- Workflow
SECUREKI
User PC with SecureKi Web
Target Devices
Use Webex Team for event notification
This document is strictly private, confidential and personal to its recipients and should not be copied, distributed or reproduced in whole or in part, nor passed to any third party without the prior written consent of Glocomp Systems (M) Sdn Bhd.
glocomp

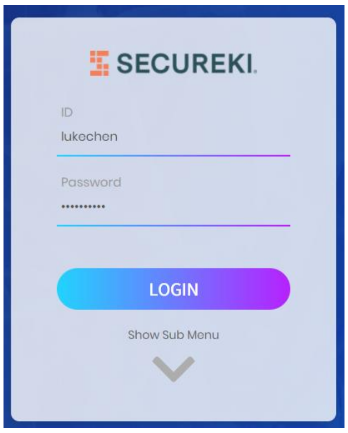
<u>API Steps:-</u>

1. Check if DUO is active (auth/v2/ping)
2. Send PUSH notification to **Cisco DUO** users
   a. Encode the password (generate from *date_authorization_header.py)*
   b. Call the POST api (auth/v2/auth)
3. Wait for the API to response.
4. Success/Fail to login to APPM
5. User access to SSH/RDP using APPMClient
6. Admin will receive event notification via **Webex Team**

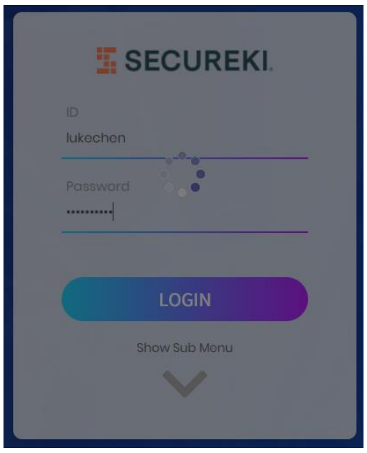
Use Case 1 – User approve the DUO push

1.1. APPM WEB

1. User login to APPM using username and password

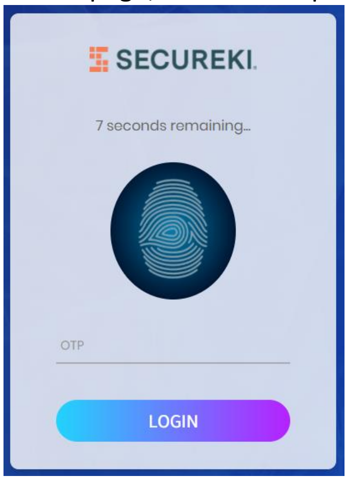


2. After user clicks the Login button, APPM should check if DUO server is up and running.
   *If stat is **OK**, proceed to step 3.*
   *If stat is **Fail**, show message "There was a problem accessing to DUO"*



GET: https://{{duo-API-HOST}}/auth/v2/ping
{"response": {"time": 1619186110}, "stat": "OK"}

3. At 2FA page, Either enter passcode or approve/reject DUO Push



Authorization

Key	Value	Remarks
Username	<To be provided>	Client ID
Password	<<encrypted string>>	Generate from *date_authorization_header.py* Format as below:- *<Date>* *<GET/POST>* *<API_ADDRESS>* *<API_URL>* *<API_QUERY>* **e.g.** Fri, 23 Apr 2021 23:59:03 +0800 POST api-*.duosecurity.com /auth/v2/auth device=auto&factor=push&username=lukechen

HEADER

Key	Value	Remarks
Date	Fri, 23 Apr 2021 23:59:03 +0800	Date format as RFC2822
Content-Type	application/x-www-form-urlencoded	

PUSH

Key	Value	Remarks
device	auto	Auto select user's device to be pushed
factor	push / auto	Get this value from Duo Radius Code If user key in "push", set factor=push
username	Secureki	Person ID

POST: `https://{{duo-API-HOST}}/auth/v2/auth?device=auto&factor=push&username=lukechen`

PASSCODE

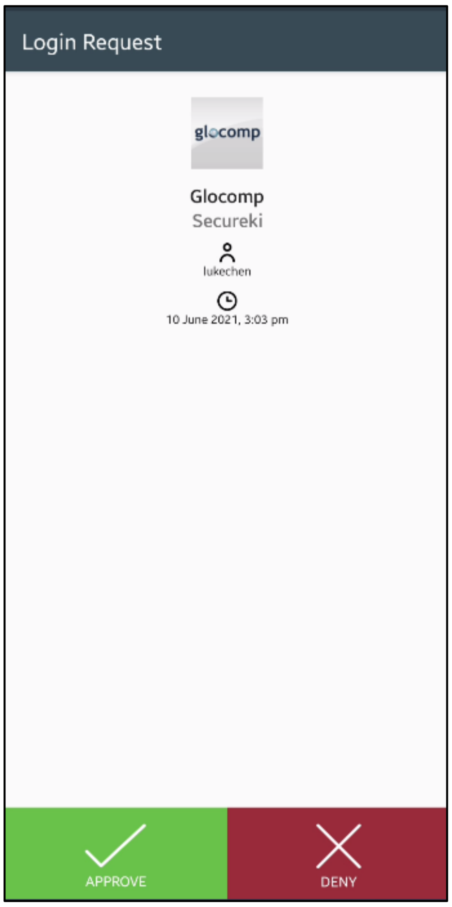
Key	Value	Remarks
factor	passcode	Get this value from **Duo Radius Code** If user key in 6-digits number, set factor=passcode
Passcode	123456	6-digits passcode
username	Secureki	Person ID

POST: `https://{{duo-API-HOST}}/auth/v2/auth?factor=passcode&passcode=123456&username=lukechen`

Response/Return:-

Status	Results
Approve	{"response": {"result": "allow", "status": "allow", "status_msg": "Success. Logging you in..."}, "stat": "OK"}
Deny	{"response": {"result": "deny", "status": "deny", "status_msg": "Login request denied."}, "stat": "OK"} {"response": {"result": "deny", "status": "fraud", "status_msg": "Login request reported as fraudulent."}, "stat": "OK"}
Timed out	{"response": {"result": "deny", "status": "timeout", "status_msg": "Login timed out."}, "stat": "OK"}
Invalid User	{"code": 40002, "message": "Invalid request parameters", "message_detail": "username", "stat": "FAIL"}

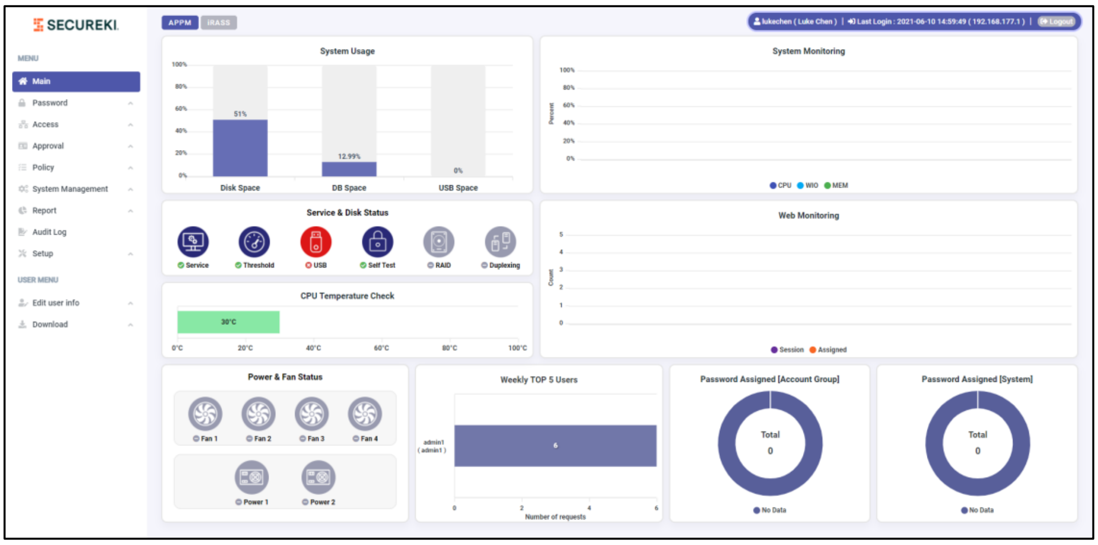
4. User will receive the **PUSH** notification to approve or deny



(Phone notification)
(duo mobile app)

5. APPM Web will wait for the user to response on their DUO mobile (**Timeout**: 1 min)

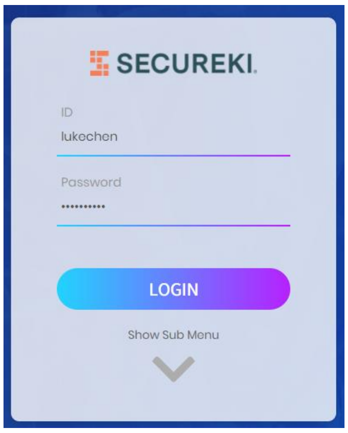
6. Once approve, user successfully login to APPM Web



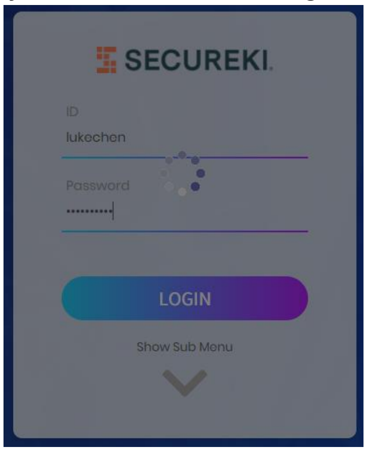
Use Case 2 – User denies the DUO push

2.1. APPM WEB

1. User login to APPM using username and password

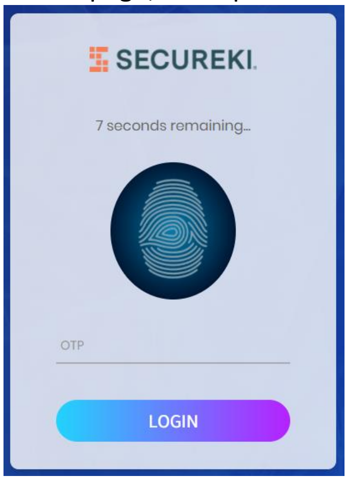


2. After user clicks the Login button, APPM should check if DUO server is up and running.
   *If stat is **OK**, proceed to step 3.*
   *If stat is **Fail**, show message "There was a problem accessing to DUO"*

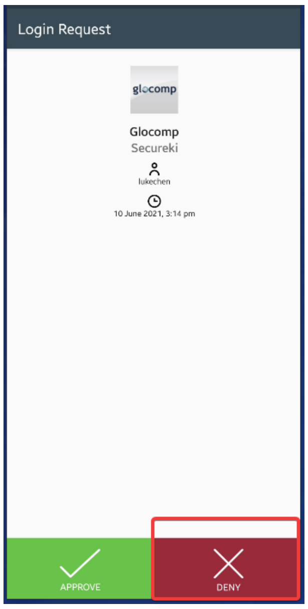


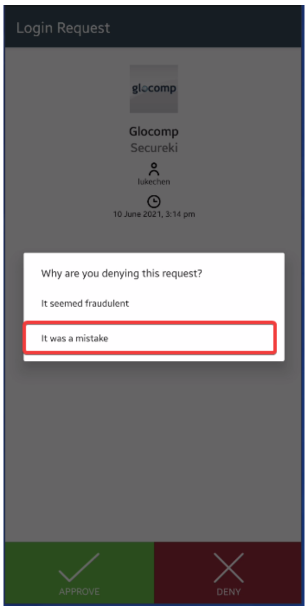
GET: https://{{duo-API-HOST}}/auth/v2/ping
{"response": {"time": 1619186110}, "stat": "OK"}

3. At 2FA page, Enter passcode or approve/deny the DUO PUSH



4. User will receive the **PUSH** notification to deny it

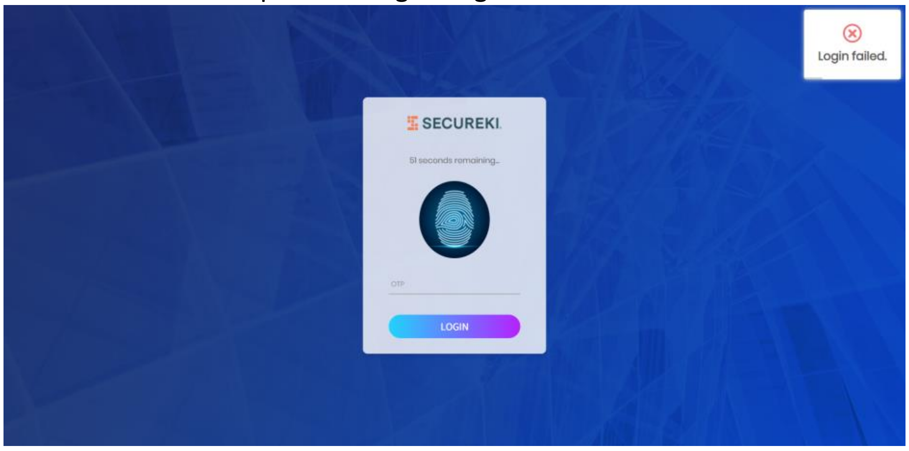




(Phone notification)
 (duo mobile app)

5. APPM Web will wait for the user to response on their DUO mobile (**Timeout**: 1 min)

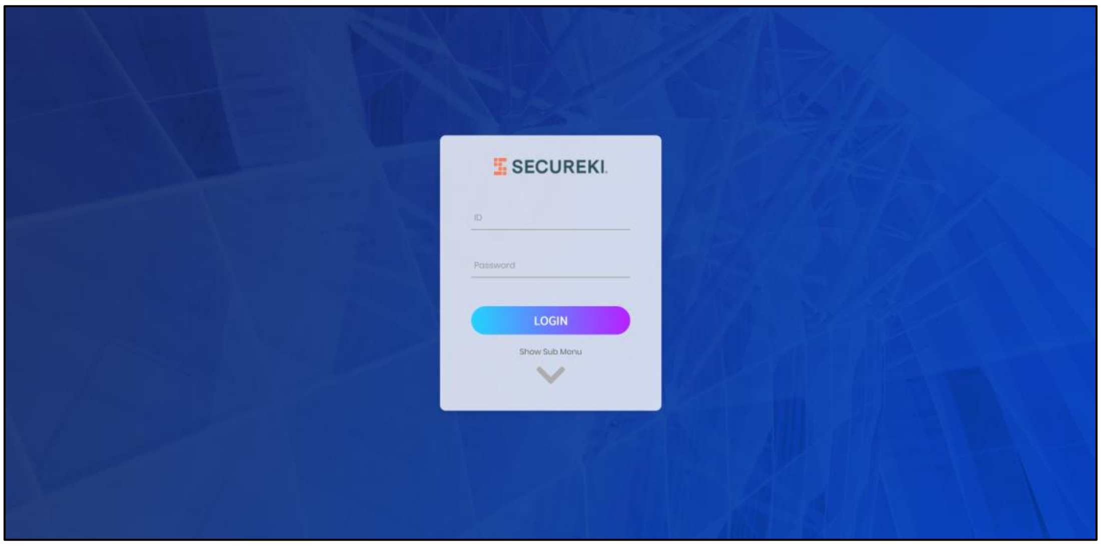
6. User denies the DUO push. User gets login failed.



Response/Return:-

Status	Results
Deny	{"response": {"result": "deny", "status": "deny", "status_msg": "Login request denied."}, "stat": "OK"} {"response": {"result": "deny", "status": "fraud", "status_msg": "Login request reported as fraudulent."}, "stat": "OK"}

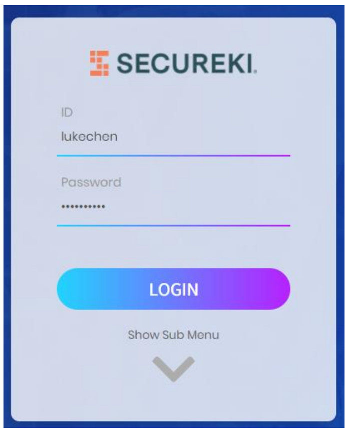
7. Redirect the APPM back to Login page



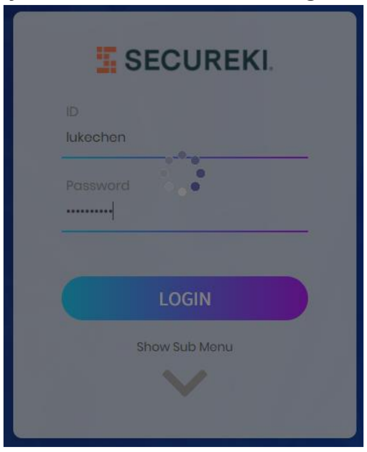
Use Case 3 – Not Valid Cisco Duo User

3.1. APPM WEB

1. User login to APPM using username and password

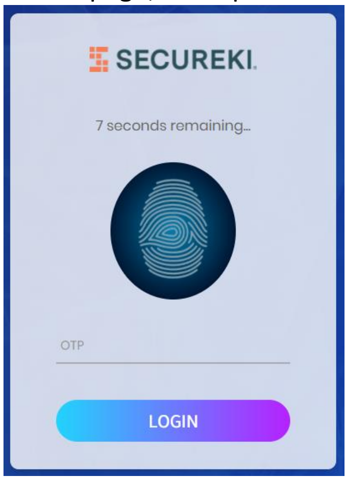


2. After user clicks the Login button, APPM should check if DUO server is up and running.
   *If stat is **OK**, proceed to step 3.*
   *If stat is **Fail**, show message "There was a problem accessing to DUO"*



GET: https://{{duo-API-HOST}}/auth/v2/ping
{"response": {"time": 1619186110}, "stat": "OK"}

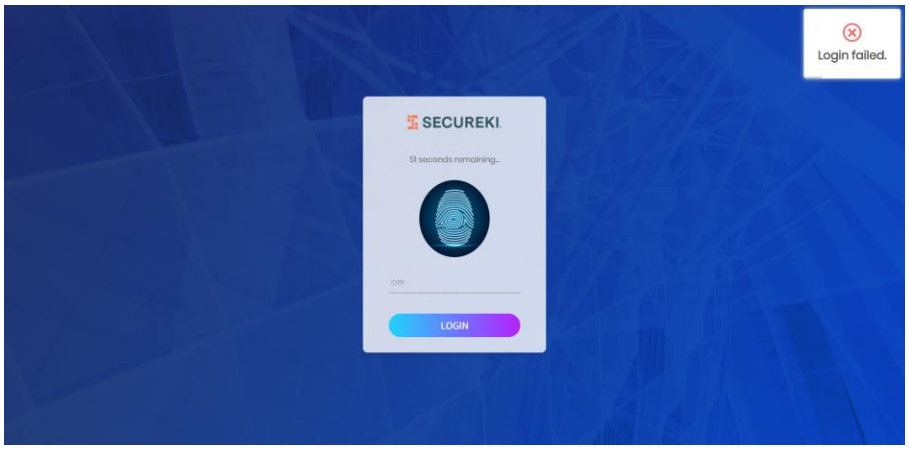
3. At 2FA page, Enter passcode or approve/deny the DUO PUSH



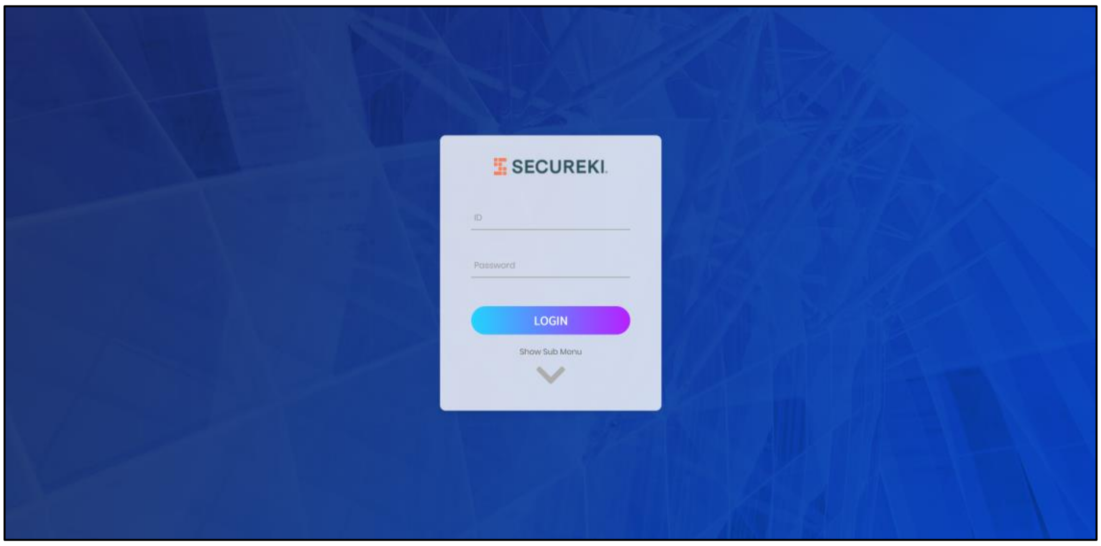
Response/Return:-

Status	Results
Invalid User	{"code": 40002, "message": "Invalid request parameters", "message_detail": "username", "stat": "FAIL"}

4. Error prompt "**Login Failed.**"



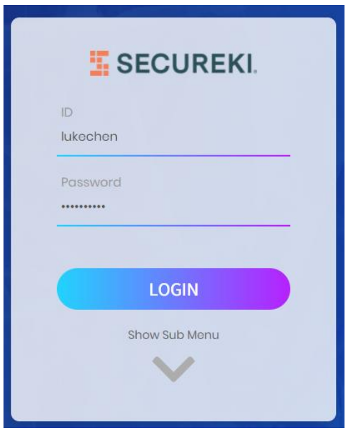
5. Redirect the user back to main login page.



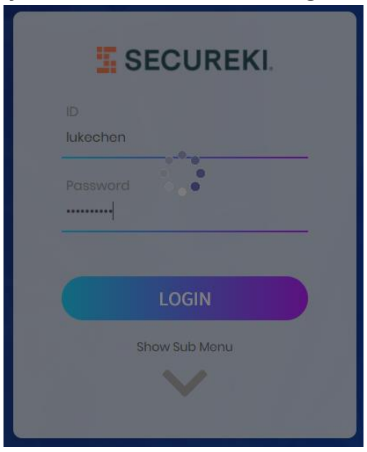
Use Case 4 – Cisco DUO is inaccessible

4.1. APPM WEB

1. User login to APPM using username and password

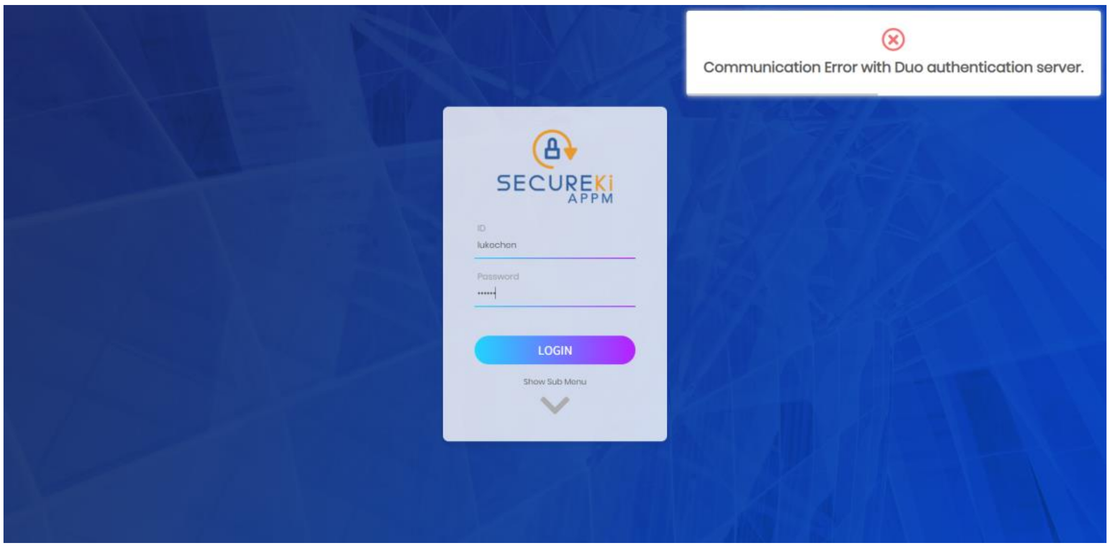


2. After user clicks the Login button, APPM should check if DUO server is up and running.
   *If stat is **OK**, proceed to step 3.*
   *If stat is **Fail**, show message "There was a problem accessing to DUO"*



   GET: https://{{duo-API-HOST}}/auth/v2/ping
   {"response": {"time": 1619186110}, "stat": "Fail"}

3. APPM login failed when ping is fail. (e.g. There was an error communicating with Duo authentication server.)

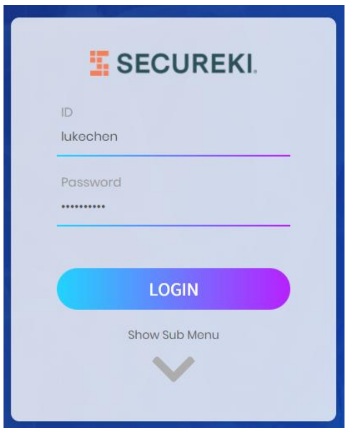


Communication Error with Duo authentication server.

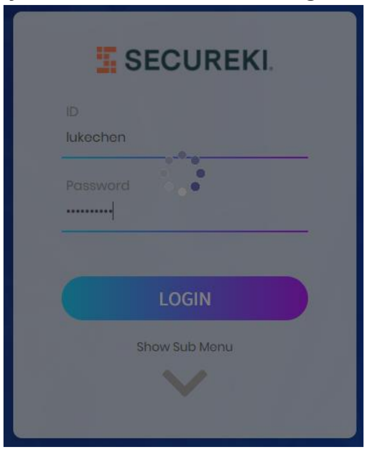
Use Case 5 – User does not response to the DUO Push / Timed Out

5.1. APPM WEB

1. User login to APPM using username and password

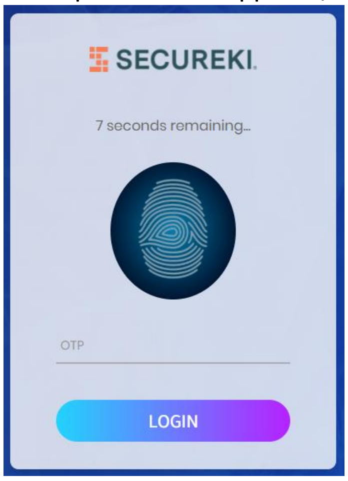


2. After user clicks the Login button, APPM should check if DUO server is up and running.
   *If stat is **OK**, proceed to step 3.*
   *If stat is **Fail**, show message "There was a problem accessing to DUO"*

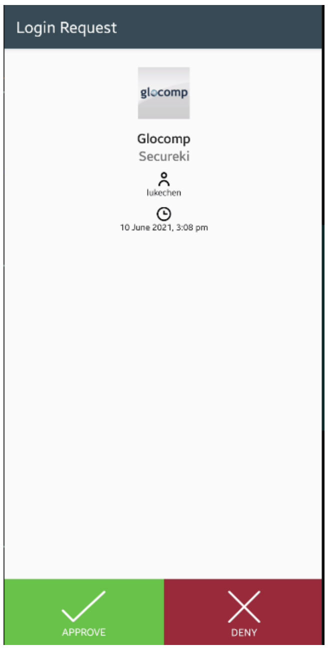


GET: https://{{duo-API-HOST}}/auth/v2/ping
{"response": {"time": 1619186110}, "stat": "OK"}

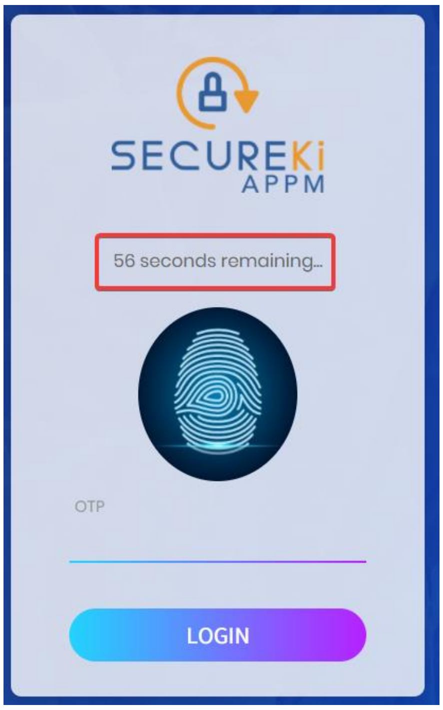
3. Enter passcode or approve/deny the DUO PUSH



4. User will receive the **PUSH** notification to approve or deny



5. APPM Web will wait for the user to response on their DUO mobile (**Timeout**: 1 min)



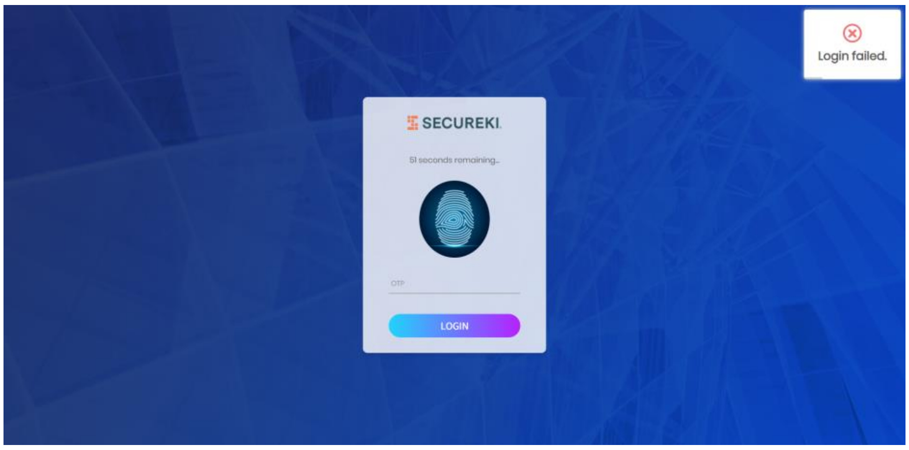
6. User does NOT approve/deny the request.

7. Request is timed out after 1 minute.

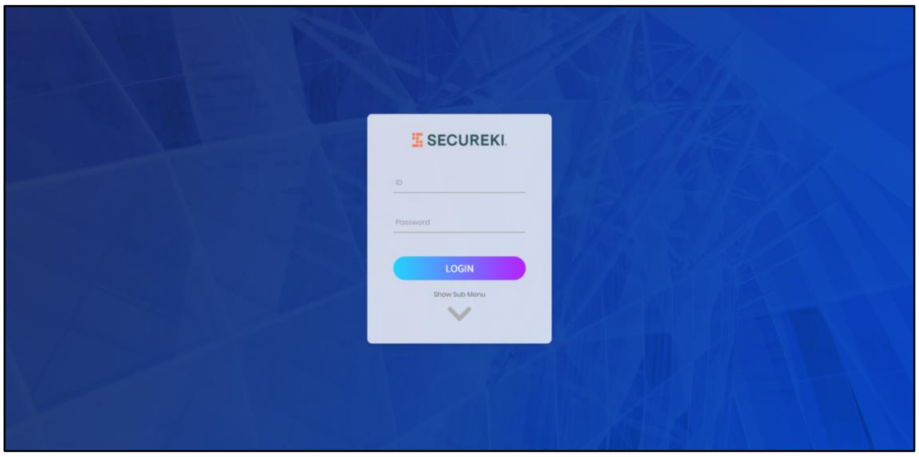
Response/Return:-

Status	Results
Timed out	{"response": {"result": "deny", "status": "timeout", "status_msg": "Login timed out."}, "stat": "OK"}

8. Error prompt using the status_msg (e.g. **Login timed out.)**



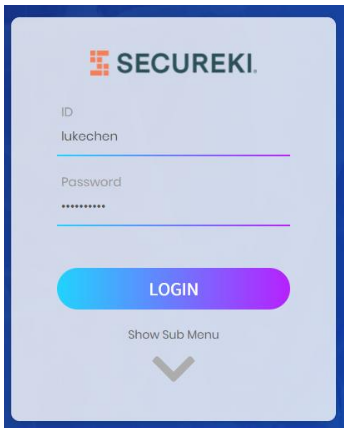
9. Redirect the APPM back to Login page



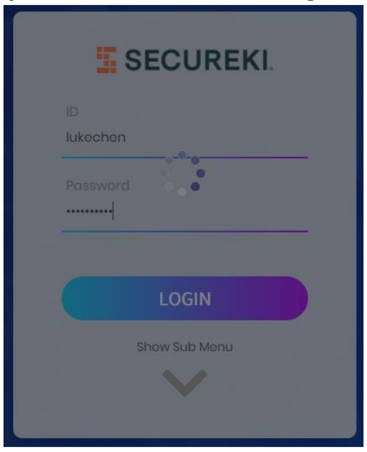
Use Case 6 – User enters '6-digits passcode' instead of 'push'

6.1. APPM WEB

1. User login to APPM using username and password

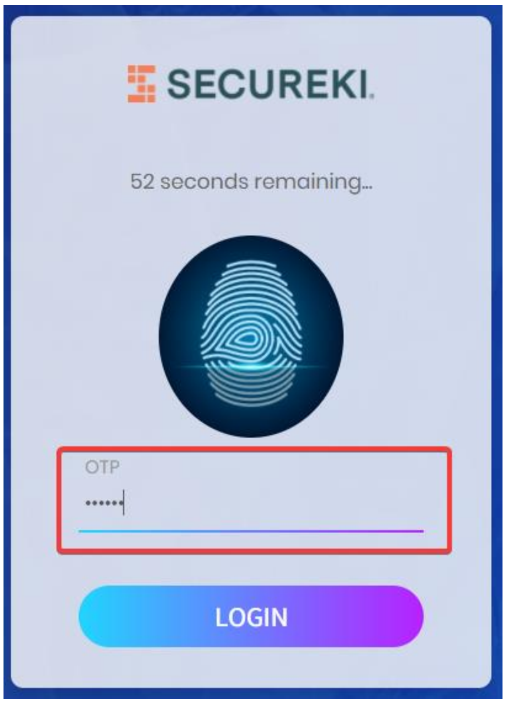


2. After user clicks the Login button, APPM should check if DUO server is up and running.
   *If stat is **OK**, proceed to step 3.*
   *If stat is **Fail**, show message "There was a problem accessing to DUO"*



GET: https://{{duo-API-HOST}}/auth/v2/ping
{"response": {"time": 1619186110}, "stat": "OK"}

3. User will type “**6-digits passcode”** and click Login



Authorization

Key	Value	Remarks
Username	<To be provided>	Client ID
Password	<<encrypted string>>	Generate from *date_authorization_header.py* Format as below:- *<Date>* *<GET/POST>* *<API_ADDRESS>* *<API_URL>* *<API_QUERY>* **e.g.** Fri, 23 Apr 2021 23:59:03 +0800 POST api-*.duosecurity.com /auth/v2/auth factor=passcode&passcode=123456&username=lukechen

HEADER

Key	Value	Remarks
Date	Fri, 23 Apr 2021 23:59:03 +0800	Date format as RFC2822
Content-Type	application/x-www-form-urlencoded	

PASSCODE

Key	Value	Remarks
factor	passcode	Get this value from **Duo Radius Code** If user key in 6-digits number, set factor=passcode
Passcode	123456	6-digits passcode
username	Secureki	Person ID

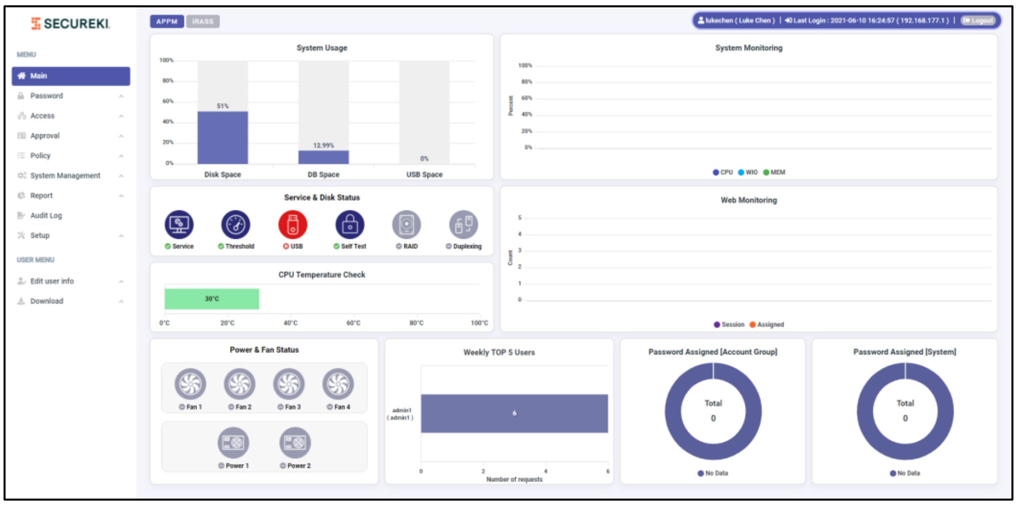
POST: `https://{{duo-API-HOST}}/auth/v2/auth?factor=passcode&passcode=123456&username=lukechen`

Response/Return:-

Status	Results
Approve	{"response": {"result": "allow", "status": "allow", "status_msg": "Success. Logging you in..."}, "stat": "OK"}
Deny	{"response": {"result": "deny", "status": "deny", "status_msg": "Login request denied."}, "stat": "OK"} {"response": {"result": "deny", "status": "fraud", "status_msg": "Login request reported as fraudulent."}, "stat": "OK"}
Timed out	{"response": {"result": "deny", "status": "timeout", "status_msg": "Login timed out."}, "stat": "OK"}
Invalid User	{"code": 40002, "message": "Invalid request parameters", "message_detail": "username", "stat": "FAIL"}

4. APPM Web will wait for the user to response on their DUO mobile (**Timeout**: 1 min)

5. Once the 6-digits passcode is valid, user successfully login to APPM Web



Use Case 7 – Notification in Webex Team via Webex API

7.1. APPM SERVICE

1. APPM Service will constantly check database for events. Once an event matched the criteria it will send a message to pre-created Webex Space via code snippet below

```
curl --request POST --url https://webexapis.com/v1/messages --header 'Authorization: Bearer ACCESS_TOKEN' --header 'Content-Type: application/json' --data '{"roomId": "ROOM_ID","text": "FORMATTED_MESSAGE"}'
```

Key	Value	Remarks
ACCESS_TOKEN	<access token>	Token is generated when creating bot at https://developer.webex.com/
ROOM_ID	<room id>	Room id retrieved via https://webexapis.com/v1/rooms
FORMATTED_MESSAGE	<pre-formatted message>	

HEADER

Key	Value	Remarks
Content-Type	application/json	

RESPONSE

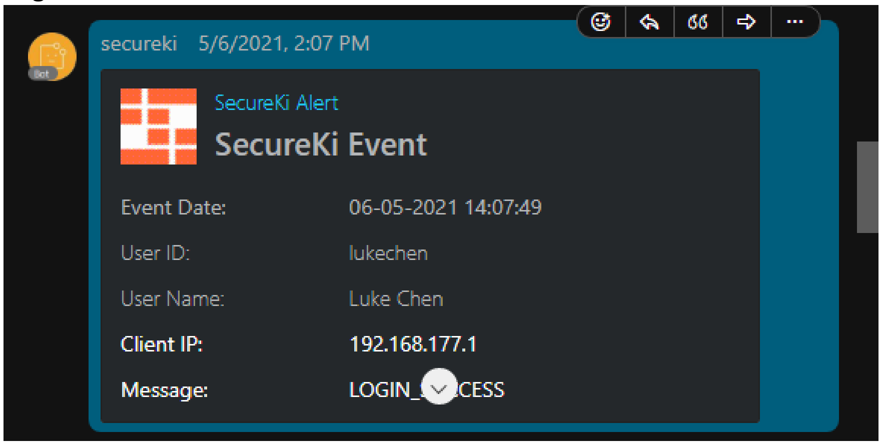
below are the response codes refer to webexapi documentation

Code	Status	Description
200	OK	Successful request with body content.
204	No Content	Successful request without body content.
400	Bad Request	The request was invalid or cannot be otherwise served. An accompanying error message will explain further.
401	Unauthorized	Authentication credentials were missing or incorrect.
403	Forbidden	The request is understood, but it has been refused or access is not allowed.
404	Not Found	The URI requested is invalid or the resource requested, such as a user, does not exist. Also returned when the requested format is not supported by the requested method.
405	Method Not Allowed	The request was made to a resource using an HTTP request method that is not supported.
409	Conflict	The request could not be processed because it conflicts with some established rule of the system. For example, a person may not be added to a room more than once.
410	Gone	The requested resource is no longer available.

Code	Status	Description
415	Unsupported Media Type	The request was made to a resource without specifying a media type or used a media type that is not supported.
423	Locked	The requested resource is temporarily unavailable. A Retry-After header may be present that specifies how many seconds you need to wait before attempting the request again.
428	Precondition Required	File(s) cannot be scanned for malware and need to be force downloaded.
429	Too Many Requests	Too many requests have been sent in a given amount of time and the request has been rate limited. A Retry-After header should be present that specifies how many seconds you need to wait before a successful request can be made.
500	Internal Server Error	Something went wrong on the server. If the issue persists, feel free to contact the Webex Developer Support team.
502	Bad Gateway	The server received an invalid response from an upstream server while processing the request. Try again later.
503	Service Unavailable	Server is overloaded with requests. Try again later.
504	Gateway Timeout	An upstream server failed to respond on time. If your query uses max parameter, please try to reduce it.

7.2. Event Notification from Webex Team Space

1. Login Success

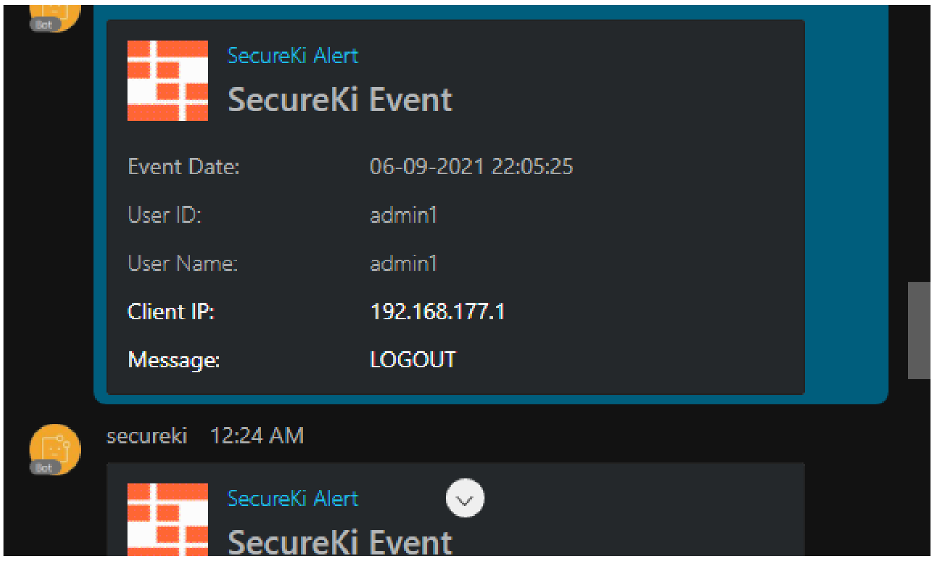

secureki 5/6/2021, 2:07 PM

SecureKi Alert

SecureKi Event

Event Date:	06-05-2021 14:07:49
User ID:	lukechen
User Name:	Luke Chen
Client IP:	192.168.177.1
Message:	LOGIN_S CESS

2. Logout



SecureKi Alert

SecureKi Event

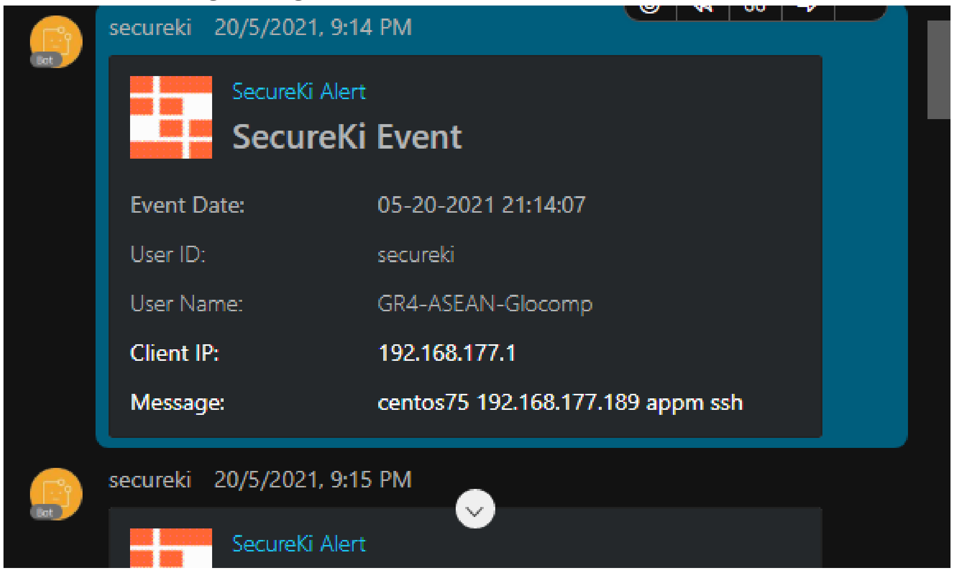
Event Date:	06-09-2021 22:05:25
User ID:	admin1
User Name:	admin1
Client IP:	192.168.177.1
Message:	LOGOUT

secureki 12:24 AM

SecureKi Alert

SecureKi Event

3. Remote accessing to target devices



secureki 20/5/2021, 9:14 PM

SecureKi Alert

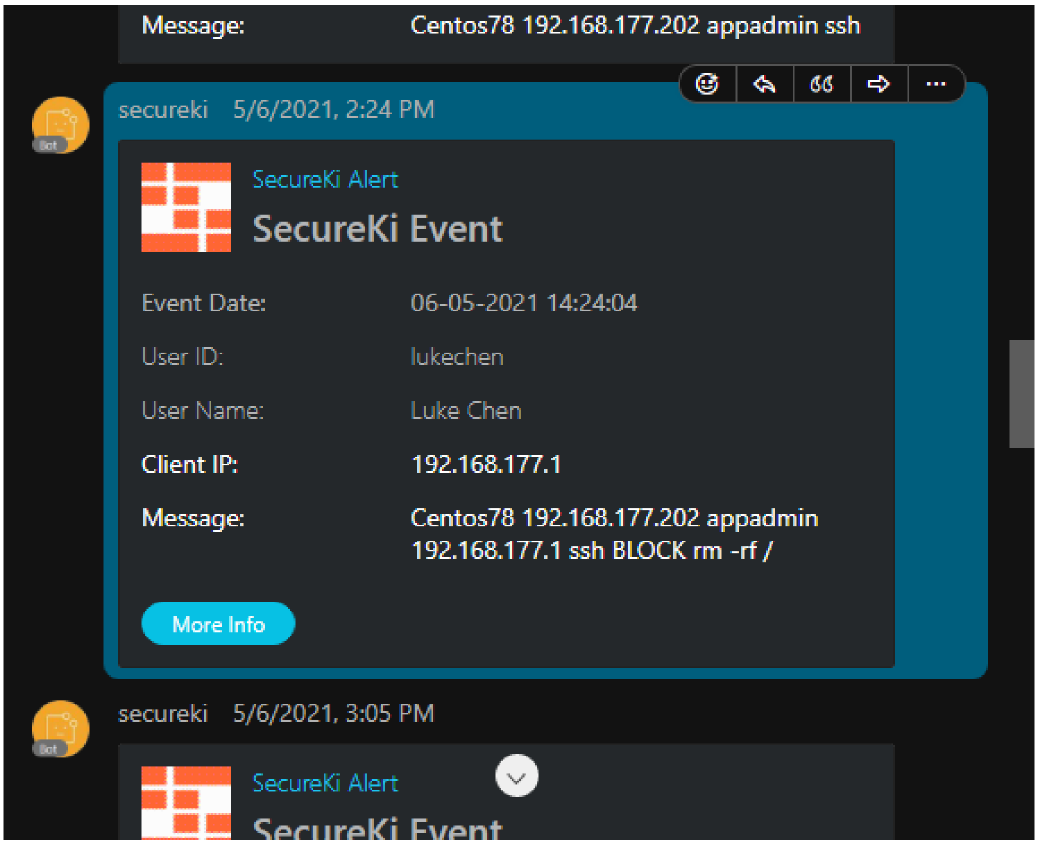
SecureKi Event

Event Date:	05-20-2021 21:14:07
User ID:	secureki
User Name:	GR4-ASEAN-Glocomp
Client IP:	192.168.177.1
Message:	centos75 192.168.177.189 appm ssh

secureki 20/5/2021, 9:15 PM

SecureKi Alert

4. Policy Violation with Card and Button



Message: Centos78 192.168.177.202 appadmin ssh

secureki 5/6/2021, 2:24 PM

SecureKi Alert

SecureKi Event

Event Date:	06-05-2021 14:24:04
User ID:	lukechen
User Name:	Luke Chen
Client IP:	192.168.177.1
Message:	Centos78 192.168.177.202 appadmin 192.168.177.1 ssh BLOCK rm -rf /

secureki 5/6/2021, 3:05 PM

SecureKi Alert

SecureKi Event

Use Case 8 – Workflow Approval via WebEx Cards and Buttons

8.1. APPM SERVICE

1. APPM Service will constantly check database for events. Once an event matched the criteria it will send a message to pre-created Webex Space via code snippet below

```
curl --request POST --url https://webexapis.com/v1/messages --header 'Authorization: Bearer
<ACCESS_TOKEN>' --header 'Content-Type: application/json' -d @<FORMATTED_MESSAGE>
```

Key	Value	Remarks
ACCESS_TOKEN	<access token>	Token is generated when creating bot at https://developer.webex.com/
FORMATTED_MESSAGE	<pre-formatted message>	

HEADER

Key	Value	Remarks
Content-Type	application/json	

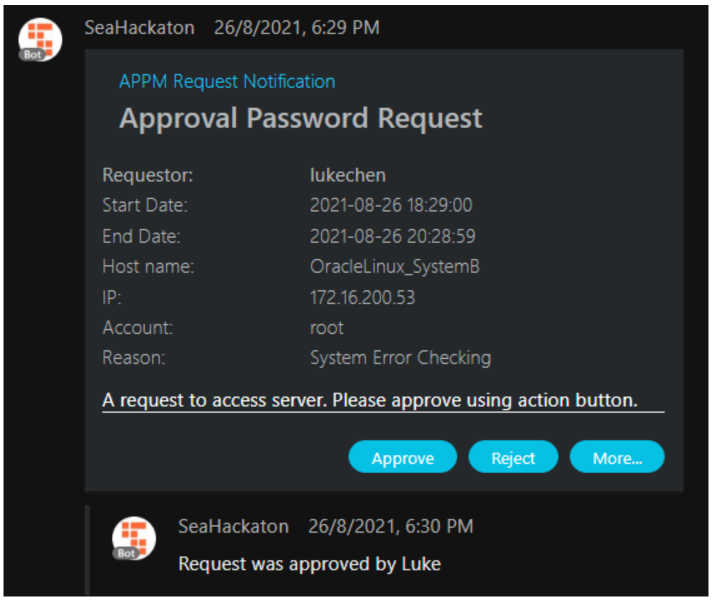
CARD DESIGN TEMPLATE

```
{
   "attachments":[
      {
         "contentType":"application/vnd.microsoft.card.adaptive",
         "content":{
            "type":"AdaptiveCard",
            "body":[
               {
                  "type":"ColumnSet",
                  "columns":[
                     {
                        "type":"Column",
                        "items":[
                           {
                              "type":"Image",

"url":"https://s3.amazonaws.com/cdn.freshdesk.com/data/helpdesk/attachments/production/62
000358262/logo/Y2UUOZXTs_blFrPk-GY93pIkeEm4c5EFLw.png",
                              "size":"Medium",
                              "height":"50px"
                           }
                        ],
                        "width":"auto"
                     },
                     {
                        "type":"Column",
                        "items":[
                           {
                              "type":"TextBlock",
               "text":"APPM Request Notification",
               "weight":"Lighter",
               "color":"Accent"
            },
            {
               "type":"TextBlock",
               "weight":"Bolder",
               "text":"Approval Password Request",
               "wrap":true,
               "color":"Light",
               "size":"Large",
               "spacing":"Small"
            }
         ],
         "width":"stretch"
      }
   ]
},
{
   "type":"ColumnSet",
   "columns":[
      {
         "type":"Column",
         "width":35,
         "items":[
            {
               "type":"TextBlock",
               "text":"Requestor:",
               "color":"Light"
            },
            {
               "type":"TextBlock",
               "text":"Start Date:",
               "weight":"Lighter",
               "color":"Light",
               "spacing":"Small"
            },
            {
               "type":"TextBlock",
               "text":"End Date:",
               "weight":"Lighter",
               "color":"Light",
               "spacing":"Small"
            },
            {
               "type":"TextBlock",
               "text":"Host name:",
               "weight":"Lighter",
               "color":"Light",
               "spacing":"Small"
            },
                {
                   "type":"TextBlock",
                   "text":"IP:",
                   "weight":"Lighter",
                   "color":"Light",
                   "spacing":"Small"
                },
                {
                   "type":"TextBlock",
                   "text":"Account:",
                   "weight":"Lighter",
                   "color":"Light",
                   "spacing":"Small"
                },
                {
                   "type":"TextBlock",
                   "text":"Reason:",
                   "weight":"Lighter",
                   "color":"Light",
                   "spacing":"Small"
                }
             ]
          },
          {
             "type":"Column",
             "width":65,
             "items":[
                {
                   "type":"TextBlock",
                   "text":"[REQUESTER_NAME]",
                   "color":"Light"
                },
                {
                   "type":"TextBlock",
                   "text":"[STRDATE]",
                   "color":"Light",
                   "weight":"Lighter",
                   "spacing":"Small"
                },
                {
                   "type":"TextBlock",
                   "text":"[ENDDATE]",
                   "weight":"Lighter",
                   "color":"Light",
                   "spacing":"Small"
                },
                {
                   "type":"TextBlock",
                   "text":"[HOSTNAME]",
                   "weight":"Lighter",
                   "color":"Light",
                  "spacing":"Small"
                },
                {
                  "type":"TextBlock",
                  "text":"[IPADDR]",
                  "weight":"Lighter",
                  "color":"Light",
                  "spacing":"Small"
                },
                {
                  "type":"TextBlock",
                  "text":"[ACCOUNTID]",
                  "weight":"Lighter",
                  "color":"Light",
                  "spacing":"Small"
                },
                {
                  "type":"TextBlock",
                  "text":"[REASON]",
                  "weight":"Lighter",
                  "color":"Light",
                  "spacing":"Small"
                }
              ]
            }
          ],
          "spacing":"Padding",
          "horizontalAlignment":"Center"
        },
        {
          "type":"TextBlock",
          "text":"A request to access server. Please approve using action button.",
          "wrap":true
        },
        {
          "type":"ActionSet",
          "actions":[
            {
              "type":"Action.Submit",
              "title":"Approve",
              "style":"positive",
              "data":{
                "action":"approve",
                "target":"[URL_APPROVE]"
              }
            },
            {
              "type":"Action.Submit",
              "title":"Reject",
              "style":"destructive",
              "data":{
```

```
                        "action":"reject",
                        "target":"[URL_REJECT]"
                      }
                    },
                    {
                      "type":"Action.OpenUrl",
                      "title":"More...",
                      "url":"https://<IPADDR>:8443/OnceApproval?[ONCE_URL]"
                    }
                  ],
                  "spacing":"None",
                  "horizontalAlignment":"Right",
                  "separator":true
                }
              ],
              "$schema":"http://adaptivecards.io/schemas/adaptive-card.json",
              "version":"1.2"
            }
          }
        ],
        "roomId":"<ROOM_TOKEN_ID>",
        "text":" "
      }
```

EXAMPLE



SeaHackaton 26/8/2021, 6:29 PM

APPM Request Notification

Approval Password Request

Requestor:	lukechen
Start Date:	2021-08-26 18:29:00
End Date:	2021-08-26 20:28:59
Host name:	OracleLinux_SystemB
IP:	172.16.200.53
Account:	root
Reason:	System Error Checking

A request to access server. Please approve using action button.

SeaHackaton 26/8/2021, 6:30 PM

Request was approved by Luke

Use Case 9 – Approved Notification via WebEx Team Space

9.1. APPM SERVICE

1. Once the request is approved, a direct message will be sent to the requester.

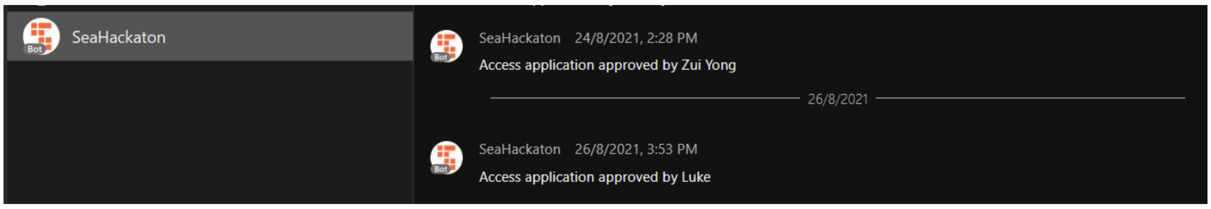
```
curl --request POST --url https://webexapis.com/v1/messages --header 'Authorization: Bearer
<ACCESS_TOKEN>' --header 'Content-Type: application/json' -d '{"toPersonEmail":
"<EMAIL_ADDR>","text": "Access application rejected by <APPROVER>"}
```

Key	Value	Remarks
ACCESS_TOKEN	<access token>	Token is generated when creating bot at https://developer.webex.com/
toPersonEmail	<EMAIL_ADDR>	Room id retrieved via https://webexapis.com/v1/rooms
text	FORMATTED_MESSAGE	Access application rejected by <APPROVER_NAME>

HEADER

Key	Value	Remarks
Content-Type	application/json	

Example



SeaHackaton

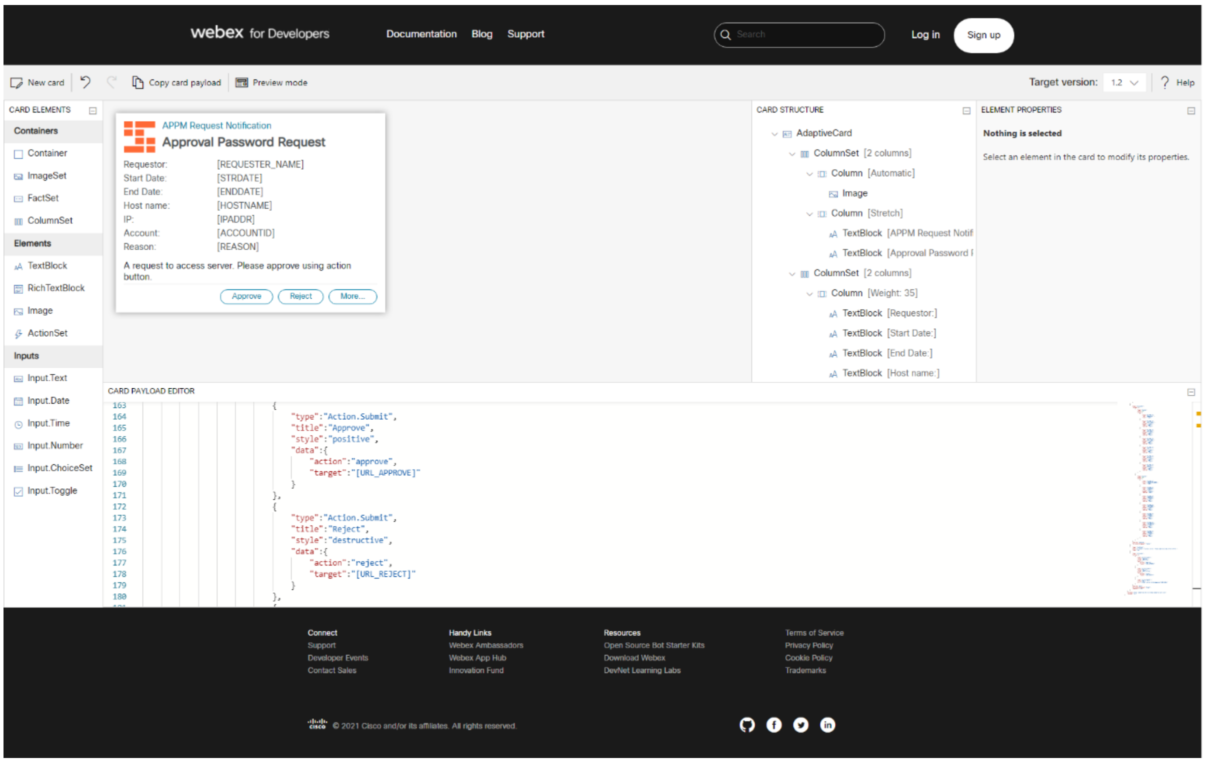
SeaHackaton 24/8/2021, 2:28 PM
Access application approved by Zui Yong

26/8/2021

SeaHackaton 26/8/2021, 3:53 PM
Access application approved by Luke

APPENDIX

1. https://developer.cisco.com/codeexchange/github/repo/hpreston/webexteamsbot/
2. https://developer.webex.com/buttons-and-cards-designer



APPM Request Notification
Approval Password Request

Requestor:	[REQUESTER_NAME]
Start Date:	[STRDATE]
End Date:	[ENDDATE]
Host name:	[HOSTNAME]
IP:	[IPADDR]
Account:	[ACCOUNTID]
Reason:	[REASON]

A request to access server. Please approve using action button.

Approve | Reject | More...

CARD STRUCTURE

```
AdaptiveCard
    ColumnSet [2 columns]
        Column [Automatic]
            Image
        Column [Stretch]
            TextBlock [APPM Request Notif
            TextBlock [Approval Password I
    ColumnSet [2 columns]
        Column [Weight: 35]
            TextBlock [Requestor:]
            TextBlock [Start Date:]
            TextBlock [End Date:]
            TextBlock [Host name:]
```

ELEMENT PROPERTIES

Nothing is selected

Select an element in the card to modify its properties.

CARD PAYLOAD EDITOR

```
{
    "type":"Action.Submit",
    "title":"Approve",
    "style":"positive",
    "data":{
        "action":"approve",
        "target":"[URL_APPROVE]"
    }
},
{
    "type":"Action.Submit",
    "title":"Reject",
    "style":"destructive",
    "data":{
        "action":"reject",
        "target":"[URL_REJECT]"
    }
},
```

END OF DOCUMENT